

Accurate Inexact Calculations of Non-Homogeneous Markov Chains

Jan Řezníček, Martin Kohlík, Hana Kubátová

Department of Digital Design, Faculty of Information Technology

Czech Technical University in Prague, Technická 9

Prague, Czech Republic

{reznija5, kohlimar, hana.kubatova}@fit.cvut.cz

Abstract—Dependability models allow calculating the rate of events leading to a hazard state – a situation, where safety of the modeled dependable system is violated, thus the system may cause material loss, serious injuries or casualties. Hierarchical dependability models allow expressing multiple redundancies made at multiple levels of a system consisting of multiple cooperating blocks. The hazard rates of the blocks are calculated independently and, when combined, they are used to calculate the hazard rate of the whole system. The independent calculations are significantly faster than the calculation of a single model composed of all models of the blocks. The paper shows a method of calculating the hazard rate of the non-homogeneous Markov chains using different homogeneous probability matrices for several hundreds small time intervals. This method will allow us to calculate the hazard rate of the non-homogeneous Markov chain very accurately compared to methods based on homogeneous Markov chains.

Index Terms—Fault tolerant systems, Hierarchical systems, Reliability, Reliability engineering.

I. INTRODUCTION

State-based dependability models (Markov chains, Petri nets, etc.) are able to model online (self-)repairing capabilities of the mission-critical systems (hot-swap modular systems, re-configurable FPGAs, etc.) easily, but the disadvantage of these models is state-explosion leading to difficulties in construction, and consequently leading to the inability to compute realistic values of dependability characteristics.

The method presented in [1] is focused on simplification of state-based dependability models and methods for easier dependability parameters computations. These models are used to create hierarchical dependability model of a system, and consequently they allow dependability parameters of large and complex systems to be calculated without the state-explosion issues.

In this paper, we want to introduce a method capable to calculate hazard (failure) rate of a system modeled by a non-homogeneous Markov chain. Such method allows us to build hierarchical dependability models without simplifications and inaccuracies meant in [1].

The proposed method is demonstrated on a case study containing multiple (up to 9) identical dependable blocks configured as an N-modular redundant system (NMR). Models of the internal block redundancy used in the study systems are used as dependability models of railway/subway interlocking equipment used in Czech Republic. The case study is used to

calculate the total hazard rate of the system and to demonstrate the dependencies of time-consumption and accuracy on the parameters of the proposed method.

The paper is organized as follows: Section II provides the theoretical background and introduces Markov chains. Section III describes the proposed method. The results are shown in Section IV and Section V concludes the paper.

II. THEORETICAL BACKGROUND

The proposed method is used to calculate the samples of failure distribution function using non-homogeneous discrete Markov chains (MCs), thus both homogeneous and non-homogeneous MCs are introduced in this section. The first part of this section also contains a formula allowing a conversion from a continuous MC to a discrete one (because the majority of the models of the dependable systems are based on continuous time, but our method is designated for discrete MCs only). The calculation of failure distribution function and its discrete samples is summarized in the last part of this section.

A. Time-Homogeneous Markov Chains

Electronic reliability design handbook MIL-HDBK-338B [2] introduces time-homogeneous MC as follows:

Markov modeling processes are stochastic processes using random variables to describe the states of the process, transition probabilities for changes of state and time or event parameters for measuring the process. A stochastic process is said to be a Markov property if the conditional probability of any future event, given any past events and the present state, is independent of the past events and depends only on the present state of the process.

There are two main types of the time-homogeneous Markov chains:

- **Continuous Time Markov Chain** is defined as the transition rate matrix Q , where the element q_{ij} defines the rate of transition from state i to state j . Each element is constant in homogeneous case, thus the probability of the transition is $p_{ij}(t) = 1 - e^{-q_{ij}t}$. The elements q_{ii} are chosen such that each row of the transition rate matrix sums to zero.
- **Discrete Time Markov Chain** is defined as the probability matrix P , where the element p_{ij} defines the

probability of transition from state i to state j . The probability does not depend on time in homogeneous case. The elements p_{ii} are chosen such that each row of the probability rate matrix sums to one.

These two types of MCs can be mutually converted. We use continuous time MC to discrete time MC conversion in our method. This conversion is based on the following equation:

$$P = I + \frac{1}{\text{delta}}Q \quad (1)$$

where Q is the transition rate matrix, P is the probability matrix, I is the identity matrix, and delta is a parameter affecting the accuracy of the conversion (the larger delta is, the more accurate is the conversion).

Our method requires the evolution of the state probabilities over time to be calculated. This evolution can be described by the Chapman-Kolmogorov equation [3]:

$$P(t + \Delta t) = P(t)P(\Delta t) \quad (2)$$

B. Non-Homogeneous Markov Chains

Both discrete time MCs and continuous time MCs can be defined using variable matrix elements. Such MCs are called non-homogeneous Markov Chains.

Most methods used to calculate the state probabilities and their evolution over time cannot be used in this case, but there are methods able to estimate the non-homogeneous MC by a homogeneous one. Two estimations are introduced in [4], [5]: *Reduction to the homogeneous process* and *Constant rate matrix between different times*.

Both methods will lead to inaccurate solutions, but the second method can be modified to use several hundreds time intervals, thus a very small error can be achieved.

We use the second method based on dividing the time interval, where the evolution of the state probabilities is calculated, to several “slices”. Each time-slice has a constant probability/transition matrix, ie. the MC is homogeneous in this time-slice.

Another method presented in [6] is focused on continuous time MCs with non-exponentially distributed parameters. The method is based on extending the MC – a single transition is split to several stages (intermediate states and transitions), thus a non-exponential distribution can be achieved. There are two main disadvantages of this method: New states have to be added to the MC and the method supports a limited set of distributions only.

C. Markov Chains and Cumulative (Failure) Distribution Function

A state i of a Markov chain is called *absorbing* if it is impossible to leave this state. Therefore, the state i is absorbing if the following condition is met:

$$p_{ii} = 1 \text{ and } p_{ij} = 0 \text{ for } i \neq j \quad (3)$$

The evolution of the absorbing state probabilities over time forms the series of the samples of the *cumulative (failure) distribution function* $F(t)$ defined as the probability in a

random trial that the random variable is not greater than t , or

$$F(t) = \int_{-\infty}^t f(t) dt \quad (4)$$

III. PROPOSED METHOD DESCRIPTION

The proposed method has two main parts:

- 1) Calculate the failure distribution function of non-homogeneous model.
- 2) Calculate the hazard rate for each time-slice (interval between two consequential samples) calculated in the previous part.

A. Calculation of the Failure Distribution Function of Non-homogeneous Model

The proposed method is based on the second approach introduced in Section II-B – it uses constant rate matrices between different times. The number of time-slices and their size can be controlled by method’s parameters.

In our case, we will calculate several hundreds of discrete samples of the failure distribution function using Chapman-Kolmogorov (C-K) equation (2) shown in Section II-B. The samples has to cover the entire lifetime of the system – typical lifetime range is ca. $[1, 10^6]$ [hours], thus we use logarithmic scale in this paper.

The correct uniform distribution of the samples should be based on a geometric progression in our case, but the C-K equation leads to an arithmetic progression (it adds a constant time-slice size Δt). Using C-K equation to calculate samples distributed geometrically would lead to extremely slow calculations, thus we use mixed distribution in this paper.

The mixed distribution is based on geometric progression on a global scale – we split the lifetime of the system to intervals $[2^n, 2^{n+1}]$ [hours], but an arithmetic progression is used inside these intervals. The number of the samples inside $[2^n, 2^{n+1}]$ [hours] is denoted as s parameter in this paper.

For example: Let the first sample be 1 [hour] and $s = 4$. Therefore the samples are as follows: $\{1, 1.25, 1.5, 1.75, 2, 2.5, 3, 3.5, 4, 5, 6, \dots, 1024, 1280, 1536, 1792, 2048, \dots\}$. We can use C-K equation easily in this case, we just need to double the difference between the consequential samples (time-slice size Δt) each time we meet value 2^n [hours].

The flowchart of the method is shown in Figure 1. The description of the method follows.

Convert transition rate matrix $Q(0)$ to probability matrix $P(dt_0)$:

The conversion is based on equation (1) shown in Section II-A. The larger the delta value is, the more accurate is the conversion (more details about the result’s accuracy can be found in Section IV).

Calculate probability matrices $P(t_{start}/s)$ and $P(t_{start})$

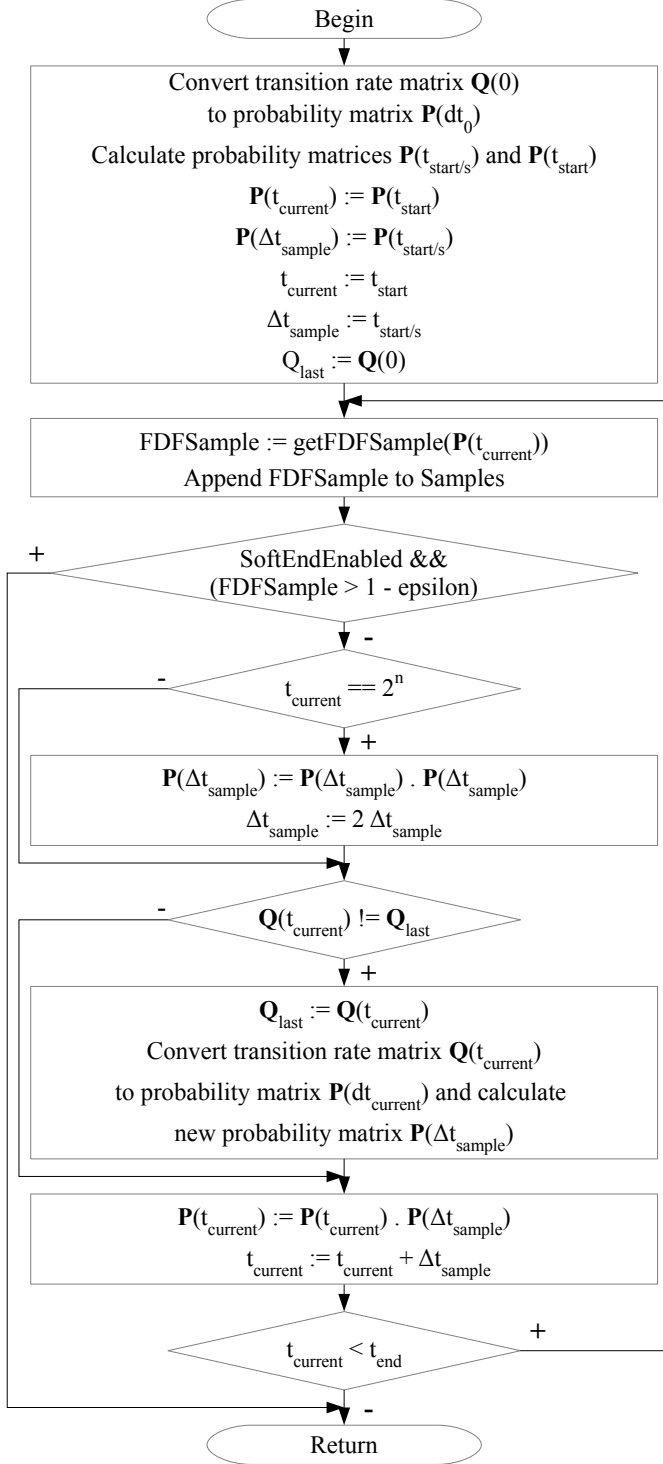


Fig. 1. Flowchart of calculation of failure distribution function of non-homogeneous model.

Both matrices are calculated from $P(dt_0)$ using a matrix power:

$$P(t_{start}/s) = P(dt_0)^{t_{start} \cdot \text{delta}/s} \quad (5)$$

$$P(t_{start}) = P(t_{start}/s)^s \quad (6)$$

Both t_{start} and s are parameters of the proposed method. t_{start} is the time, where the first sample of the failure distribution function $F(t)$ is stored. We assume that the start of $F(t)$ function is not interesting in most cases, i.e. no one needs a detailed information about failure probability during the first hour (or first ten, or first hundred hours, etc.) of operation.

The s parameter has been already introduced – it determines the number of the time-slices (samples stored) in interval $[2^n, 2^{n+1}]$ [hours]. The higher the value of s is, the more often the $F(t)$ is sampled and the calculation's execution time is increased.

getFDFSsample(P(t_{current})) function

This function calculates the value of the failure distribution function $F(t_{current})$ as the sum of the probabilities of the absorbing states.

$$\text{getFDFSsample} = \sum_{j \in \text{AbsorbingStates}} p_{1j} \quad (7)$$

assuming the first state is the initial state of the Markov chain ($p_1(0) = 1$).

The calculated sample is stored to the array of samples (*Samples*).

SoftEndEnabled && (FDFSsample > 1 - epsilon) block

We assume that the end of $F(t)$ function is not interesting in most cases, i.e. no one needs a detailed information about failure probability when it exceeds 99% (or 99.9%, etc.). This “soft” end of calculation can be parametrized using *epsilon* value or disabled completely.

Adjusting of the size of the interval between consequential samples

The size of the interval between consequential samples must be adjusted (doubled) in order to keep the mixed geometric-arithmetic progression of the samples. This adjustment is made when the current time $t_{current}$ meets value 2^n , where $n \in \mathbb{N}$.

Q(t_{current}) ≠ Q_{last} block

The main feature of non-homogeneous Markov chain is the possibility to change the rates of the events over time. If such change is detected, new $P(dt_{current})$ and $P(\Delta t_{sample})$ matrices has to be calculated using (1) and (5) respectively.

t_{current} < t_{end} block

The “hard” end of the calculation is determined by t_{end} parameter – the calculation will end here no matter to the value of $F(t)$ function.

Parameters summary

The summary of meaning of all parameters of the method is shown in Figure 2. The plot shows an example failure distribution function $F(t) = 1 - e^{(-10^{-5} \cdot t)}$ and the following parameter setting:

- $t_{start} = 2^7 = 128$ [hours]
- $t_{end} = 2^{22} \doteq 4 \times 10^6$ [hours]
- $s = 4$ [-]
- $SoftEndEnabled = false$
- $epsilon = 10^{-3}$ [-] (disabled by $SoftEndEnabled$)

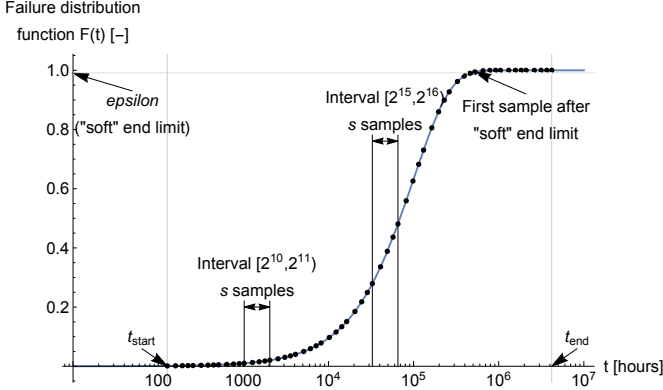


Fig. 2. Summary of meaning of all parameters on illustrative example of failure distribution function.

As you can see, the samples are stored in interval $[t_{start}, t_{end}]$ only. Each interval $[2^n, 2^{n+1})$ contains the same number of samples (s). “Soft” end is not used in this example, it would disable the calculation of the samples beyond the marked one.

The result of this part is an array containing samples of failure distribution function that can be used to calculate the hazard rate.

B. Calculate the Hazard Rate for Each Time-slice

We assume that the hazard rate is constant in the duration of the time-slice (ie. between two consequential samples of $F(t)$). The formula used on the hazard rate calculation is derived from the most simple continuous time Markov chain. This MC contains an initial state S_0 , an absorbing state S_1 and one arc leading from the initial state to the absorbing one (with rate λ). The following system of differential equations describes this MC:

$$p'_{S_0}(t) = -p_{S_0}(t) \lambda \quad (8)$$

$$p'_{S_1}(t) = p_{S_0}(t) \lambda \quad (9)$$

$$p_{S_0}(0) = 1 \quad (10)$$

$$p_{S_1}(0) = 0 \quad (11)$$

For a homogeneous case, the result $F(t)$ function of this system is

$$F(t) = p_{S_1}(t) = 1 - e^{(-\lambda \cdot t)} \quad (12)$$

Now we introduce a point t_1 , calculate the value of $p_{S_0}(t_1)$ and $p_{S_1}(t_1)$ and use these values as the initial condition of the system (8)-(11).

After this modification, we receive a non-homogeneous system made of two homogeneous intervals ($[0, t_1)$ and $[t_1, \infty)$). This modification can be used repeatedly to obtain a non-homogeneous system made of n homogeneous intervals and the following failure distribution function:

$$F(t) = \begin{cases} 1 - e^{(-\lambda_0 \cdot t)} & 0 \leq t < t_1 \\ 1 - e^{(-\lambda_1 \cdot t + \lambda_1 \cdot t_1 - \lambda_0 \cdot t_1)} & t_1 \leq t < t_2 \\ \vdots & \\ 1 - e^{(-\lambda_i \cdot t + \sum_{j=1}^i (\lambda_j \cdot t_j) - \sum_{j=1}^i (\lambda_{j-1} \cdot t_j))} & t_i \leq t < t_{i+1} \\ \vdots & \end{cases} \quad (13)$$

Both sums in the exponent are constant in the i -th interval (they do not depend on time t), thus we replaced them with a single constant denoted as *offset*.

$$F_i(t) = 1 - e^{(-\lambda_i \cdot t + offset)} \quad (14)$$

Now we take two samples bounding the i -th interval calculated from the previous part stored as pairs $\{t_i, F(t_i)\}$ and $\{t_{i+1}, F(t_{i+1})\}$ and simply create a system of two equations:

$$F(t_i) = 1 - e^{(-\lambda_i \cdot t_i + offset)} \quad (15)$$

$$F(t_{i+1}) = 1 - e^{(-\lambda_i \cdot t_{i+1} + offset)} \quad (16)$$

Assuming $t_i \neq t_{i+1}$ and $F(t_i) < F(t_{i+1}) < 1$, there is only one solution of this system:

$$\lambda_i = \frac{\log_e(1 - F(t_i)) - \log_e(1 - F(t_{i+1}))}{t_{i+1} - t_i} \quad (17)$$

Now lets assume a general MC with a sampled $F_{general}(t)$ function calculated from the first part of the proposed method. We want the hazard rate between two consequential samples to be constant. The only MC with a constant hazard rate is the simplest MC, thus we reduce the general MC to the simplest one and use hazard rates λ_i from (17). The $F_{simple}(t)$ and $F_{general}(t)$ will not be equal in general, but they will have the same value for each calculated sample.

This simplification will introduce an inaccuracy to our solution, but we can use as many samples as we need to obtain a suitable solution.

IV. RESULTS

A. Case Study Description

The proposed method is demonstrated on a case study containing multiple (up to 9) identical dependable blocks configured as an N-modular redundant system (NMR). Models of the internal block redundancy used in the study systems are used as dependability models of railway/subway interlocking

equipment used in Czech Republic. The case study is used to demonstrate the dependencies of time-consumption and accuracy on the parameters of the proposed method. Wolfram Mathematica [7] tool is used to perform the calculations.

The dependable blocks used in the case study system use Two-out-of-two (2oo2) redundancy [8], [9]. Each dependable block contains two independent copies of functional modules, thus the safety of the blocks using these redundancies cannot be violated by a single fault. The detailed description of the system and its model follows and can be found in [1].

The model shown in Fig. 3 is used to calculate the failure distribution function $F(t)$ of the 2oo2 block.

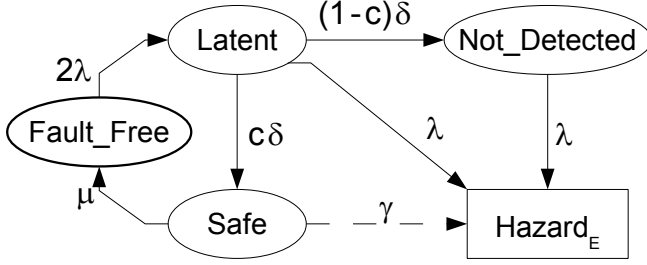


Fig. 3. Dependability model of Two-out-of-two block.

Fault_Free is the functional/fault-free state of the block. The fault rate of the first fault is 2λ , because the first fault can affect any of the two functional modules of the block.

The *Latent* state is active when the block contains a fault that has not been detected yet. The rate of the on-line test (the inverted average delay between fault origin and detection) is labeled as δ . If the test is performed successfully (a fault is detected), the block will be locked in the *Safe* state. The probability of a successful test is labeled as c .

If the test fails (the fault is present, but not detected), the block will be in the *Not_Detected* state. The safety of the block is not violated in this state, but another fault (with a fault rate λ) affecting the unaffected functional module will lead to safety violation (*Hazard_E* state). The second fault hit inside already affected functional module cannot cause a hazard, because the other functional module works correctly.

The arc leading from *Latent* to *Hazard_E* expresses the probability that a second fault will affect the unaffected functional module before the test is finished.

The block locked in the *Safe* state waits until the repair is finished (repair rate μ). The block is not functional in this state, but the safety is not violated.

The functionality of the block will be provided by a backup/emergency method (by a human operator in this case), when the block is locked in the *Safe* state. The rate γ expresses the hazard rate of the backup/emergency method (a mistake of a human operator). This rate should be included into the safety analysis if a more complex analysis needs to be done.

The probability of detection of a fault, the fault rate, and the block-lock rate of the block form the following parameters values. The values have been taken from [9].

$\mu = 24^{-1} [h^{-1}]$ – the repair rate

$\lambda = 10^{-5} [h^{-1}]$ – the fault rate

$\delta = 10^{-1} [h^{-1}]$ – the block-lock rate

$c = 0.6$ – the probability of detecting a fault by the block-lock

$\gamma = 10^{-3} [h^{-1}]$ – the backup/emergency method hazard rate

N-modular Redundancy (NMR) is based on N identical blocks and a voter. This voter is able to compare all outputs of the blocks. It uses majority voting to produce a single output. If less than half of the blocks fail, the voter is able to produce correct output. If more than half of the blocks fail, the voter will produce an incorrect output – this situation is considered as a hazard state. The erroneous blocks cannot be identified, thus there is no restoration/repair possibility.

The model shown in Fig. 4 is used to calculate the failure distribution function of a generic NMR system. The NMR system containing N blocks will contain $\lfloor \frac{N}{2} \rfloor$ transient states. These states correspond to the blocks that are in the hazard state. NMR systems consisting of 3 to 9 blocks are used in this paper.

B. Method Parameters Impact

The 3-MR (TMR) system based on 2oo2 blocks is used to demonstrate the impact of method parameters. The model of this system is made as the Cartesian product of the dependability models of three identical 2oo2 blocks and the model of the TMR. The model contains 34 states.

Two parameters that determine the trade-off between method accuracy and time-consumption were introduced in Section III:

- 1) *delta* – this parameter determines the size of the time interval used in the continuous time MC to discrete time MC conversion.
- 2) *s* – this parameter determines the number of the time-slices (samples stored) in interval $[2^n, 2^{n+1}]$ [hours].

The other parameters values are as follows:

$t_{start} = 1$ [hour] – time, where the first sample of the failure distribution function $F(t)$ is stored

$t_{end} = 2^{20} \doteq 10^6$ [hours] – time of the “hard” end

SoftEndEnabled = false

delta parameter impact

Table I shows the impact of the *delta* parameter. The size of *delta* is shown in the first column, the CPU-time¹ spent on conversion from the continuous time MC to discrete time MC and calculation of the probability matrix $P(t_{start})$ (*Initialization* time) is shown in the second column. The third column shows CPU-time spent on the rest of the calculation (*Main loop* time). The relative error of the first sample (compared to the sample taken from $delta = 2^{65}$ setting) is shown in the third column – this error is the highest of all samples

¹Running on Intel Core i5-7300HQ @2.5 GHz, OS: Win10 64-bit, Mathematica 11.2.

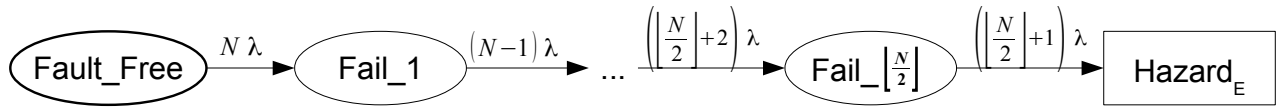


Fig. 4. Dependability model of generic N-modular redundant system.

in this case. The average of the absolute values of relative errors of all stored samples (compared to the samples taken from $\delta = 2^{65}$ setting) is shown in the last column. The absolute values are used, since there are both positive and negative values of relative errors.

The higher is the δ value, the lower is the relative error. The *Initialization* time is slightly increasing when the δ value increases, and the CPU-time spent on the *Main loop* remains nearly constant² (no additional conversion is made in *Main loop*, because the values of the parameters remain constant in this case).

TABLE I
COMPARISON OF CPU-TIMES AND RELATIVE ERRORS OF THE METHOD WITH RESPECT TO THE δ PARAMETER.

δ [-]	Initialization time [s]	Main loop time [s]	Rel. error of the first sample [-]	Average rel. error [-]
2^{10}	0.047	0.959	-1.11×10^{-2}	8.17×10^{-4}
2^{15}	0.076	0.976	-3.48×10^{-4}	2.56×10^{-5}
2^{20}	0.100	0.963	-1.09×10^{-5}	8.00×10^{-7}
2^{25} ¹⁾	0.131	0.976	-3.40×10^{-7}	2.50×10^{-8}
2^{30}	0.160	0.962	-1.06×10^{-8}	7.81×10^{-10}
2^{35}	0.187	0.950	-3.32×10^{-10}	2.44×10^{-11}
2^{40}	0.220	0.979	-1.04×10^{-11}	7.63×10^{-13}
2^{45}	0.240	0.960	-3.25×10^{-13}	2.38×10^{-14}
2^{50}	0.268	0.966	-1.01×10^{-14}	7.45×10^{-16}
2^{55}	0.290	0.980	-3.17×10^{-16}	2.33×10^{-17}
2^{60}	0.310	0.945	-9.60×10^{-18}	7.05×10^{-19}
2^{65}	0.340	0.954	—	—

¹⁾ Default value used in this paper.

s parameter impact

Table II shows the impact of the s parameter. The size of s is shown in the first column, the *Initialization* CPU-time spent on is shown in the second column. The third column shows *Main loop* CPU-time. The values of the samples are equal for all settings in this case, thus there are no relative errors in this table.

The *Initialization* time is nearly constant (the same initialization is made in all cases), but the CPU-time spent on the *Main loop* is increasing linearly with respect to the numbers of samples calculated (s parameter).

C. Time Non-Homogeneity Impact

The results presented in the previous sections were based on homogeneous case (all rates were constant all the time).

²⁾The measured CPU-times varies by ca. $\pm 10\%$, when the calculation is performed repeatedly, due to inaccurate time values provided by Mathematica.

TABLE II
COMPARISON OF CPU-TIMES OF THE METHOD WITH RESPECT TO THE s PARAMETER.

s [-]	Initialization time [s]	Main loop time [s]
2^1	0.130	0.210
2^2	0.130	0.330
2^3	0.130	0.560
2^4 ¹⁾	0.130	0.962
2^5	0.125	1.840
...		
2^{10}	0.126	57.40

¹⁾ Default value used in this paper.

The following case shows the time-consumption and relative error of the non-homogeneous case. We use the same setting as presented in the previous case, but we disable the check in the $Q(t_{current}) \neq Q_{last}$ block (see the flowchart of the method shown in Figure 1 in Section III). This modification forces the method to calculate new $P(dt_{current})$ and $P(\Delta t_{sample})$ matrices in each time-slice. The results should be equal to the homogeneous case and the time-consumption will be similar to a general non-homogeneous case.

Table III shows the impact of this modification. The type of the method is shown in the first column, the CPU-time spent on *Initialization* is shown in the second column. The third column shows *Main loop* CPU-time. The average of the absolute values of relative errors of all stored samples (compared to the samples taken from homogeneous case) is shown in the last column.

TABLE III
COMPARISON OF CPU-TIMES OF HOMOGENEOUS AND NON-HOMOGENEOUS CASE.

Case type	Initialization time [s]	Main loop time [s]	Average rel. error [-]
Homogeneous	0.130	1.030	—
Non-Homogeneous	0.141	17.83	7.20×10^{-8}

The plot shown in Fig. 5 shows the relative errors of all samples between the homogeneous and the non-homogeneous case. The horizontal axis represents the time of operation measured in hours, the vertical axis represents the size of the relative error.

As you can see, the maximal value of the relative error of non-homogeneous case is ca. 10^{-7} . The *Initialization* times are nearly equal in both cases, but the *Main loop* time has

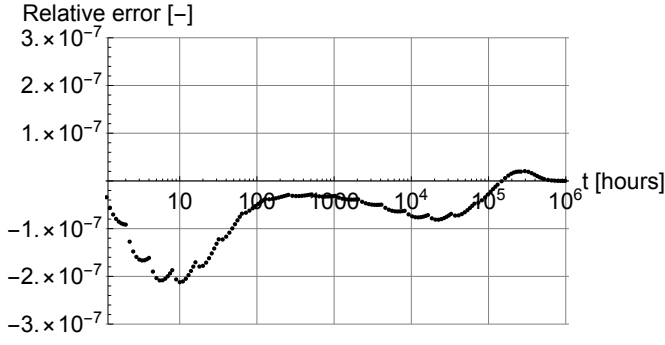


Fig. 5. Relative errors of all samples between homogeneous and non-homogeneous case.

increased ca. 20 times due to calculation of new $P(dt_{current})$ and $P(\Delta t_{sample})$ matrices in each time-slice.

D. Comparison to Analytical Solution

We use the NMR model to compare the presented method to the analytical solution in this section. The failure distribution function of the NMR system can be calculated using the following equation

$$F_{NMR}(t) = 1 - \sum_{i=M}^N \binom{N}{i} F(t)^{N-i} (1 - F(t))^i \quad (18)$$

where $F(t)$ is the failure distribution function of the single block ($F(t) = 1 - e^{(-\lambda \cdot t)}$ and $\lambda = 10^{-5}$ in this case), N is (odd) number of blocks used in the system, and M is number of blocks required to be operational ($\frac{N+1}{2}$ in this case).

Table IV shows the relative errors between the presented method and the analytical solutions. The first column shows the number of the NMR blocks, the second column shows the relative error of the first sample of the proposed method (compared to the sample taken from the analytical solution). The average of the absolute values of relative errors of all stored samples is shown in the last column.

TABLE IV
COMPARISON OF RELATIVE ERRORS OF PRESENTED METHOD AND ANALYTICAL SOLUTION.

NMR blocks	Rel. error of the first sample [-]	Average rel. error [-]
3-MR	-2.98×10^{-8}	1.69×10^{-8}
5-MR	-8.94×10^{-8}	4.92×10^{-8}
7-MR	-1.79×10^{-7}	9.76×10^{-8}
9-MR	-2.98×10^{-7}	1.62×10^{-7}

The plot shown in Fig. 6 shows the relative errors of the samples between the presented method and the analytical solution. The horizontal axis represents the time of operation measured in hours, the vertical axis represents the size of the relative error. Please note that only each third sample is

shown – the plot would be hard to read when all samples were present.

The maximal value of the relative error is slowly increasing with increasing number of the blocks, but it remains very low in all presented cases (ca. 10^{-7}).

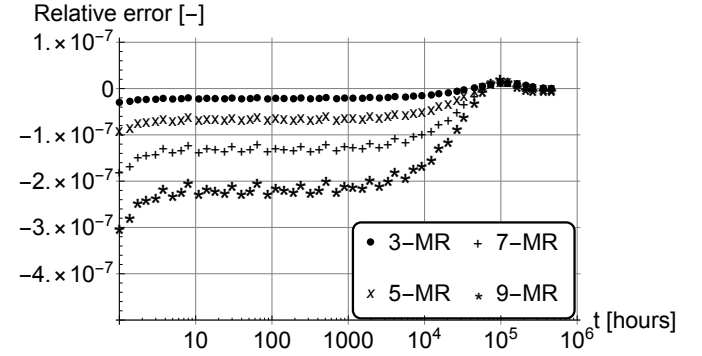


Fig. 6. Relative errors of samples between presented method and analytical solution.

E. Application on Hierarchical Model

The hierarchical dependability model of the NMR system based on 2oo2 blocks is used in this case. The model of a 2oo2 block is created, the samples of the $F(t)$ function are calculated, and the hazard (failure) rate result is taken as the fault rate (λ) of the NMR model.

The results of the hierarchical approach are compared to the results of the exact model (the model generated by the Cartesian product of the dependability models of the 2oo2 blocks and the model of the NMR).

Table V shows the comparison of the CPU-times and the relative errors of the hierarchical and the exact solutions. The first column shows the number of the 2oo2 blocks, the second column shows the CPU-time spent on exact model solution (sum of *Initialization* and *Main loop* CPU-times). The CPU-time spent on hierarchical method (*Initialization* and *Main loop* CPU-times for the 2oo2 model and the same times spent on the NMR model) is shown in the third column. The relative error of the first sample (compared to the sample taken from the exact case) is shown in the third column. The average of the absolute values of relative errors of all stored samples is shown in the last column.

TABLE V
COMPARISON OF CPU-TIMES OF EXACT AND HIERARCHICAL CASE.

NMR blocks	Exact time [s]	Hierarchical time [s]	Rel. error of the first sample [-]	Average rel. error [-]
n03	1.160	0.091	1.77×10^{-7}	2.94×10^{-8}
n05	37.23	0.130	5.31×10^{-7}	8.67×10^{-8}
n07	627.5	0.169	1.06×10^{-6}	1.73×10^{-7}
n09	5,938	0.225	1.77×10^{-6}	2.85×10^{-7}

The plot shown in Fig. 7 shows the relative errors of the samples between the exact and the hierarchical case. The

horizontal axis represents the time of operation measured in hours, the vertical axis represents the size of the relative error. Please note that only each third sample is shown – the plot would be hard to read when all samples were present.

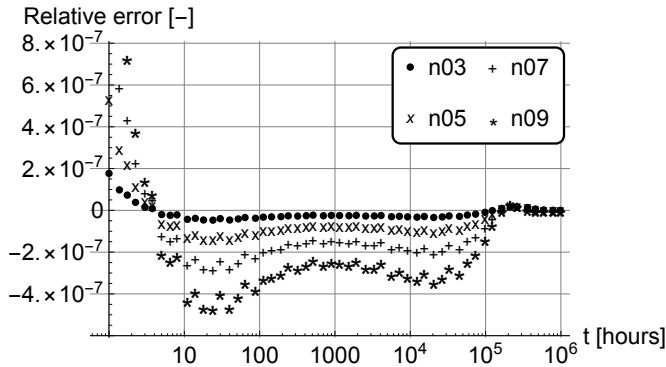


Fig. 7. Relative errors of samples between exact and hierarchical case.

The CPU-time spent on solving the exact model solution grows rapidly with increasing number of the blocks, but the CPU-time spent on hierarchical method is below 1 second in all presented cases.

The maximal value of the relative error is slowly increasing with increasing number of the blocks, but it remains very low in all presented cases (ca. 10^{-7}).

V. CONCLUSIONS

This paper presents a method able to calculate the samples of failure distribution function $F(t)$ from a non-homogeneous Markov chain. The proposed method is very accurate (the average relative error between the samples of failure distribution function $F(t)$ is ca. 10^{-8} using default parameters in the presented case studies), but it is slower (ca. 20 times in the tested case study) than the common method intended for homogeneous case.

The trade-off between relative error and the calculation time can be controlled by the method's parameters. The default parameter's setting leads to relative error ca. 10^{-8} , but the relative error can be decreased and the execution time will be increased by ca. 5 % (in homogeneous case) per an order of magnitude of the relative error.

The proposed method can be used to calculate a hazard (failure) rate λ of a hierarchical Markov chain. The method allows accurate calculations even in the case of large complex systems, where the results of classical models are practically unreachable due to state explosion. The calculation time of the classical models increases exponentially with respect to the number of dependable blocks (the Cartesian product of all models has to be made in such case), but the proposed method allows us to use the hierarchical Markov chain with small relative error and calculate each model of dependable block independently.

The samples of the non-homogeneous hazard (failure) rate of the dependable block are calculated using the samples of failure distribution function $F(t)$ and a simple equation. The

samples of the non-homogeneous hazard (failure) rate can be used as the parameter of the top-level model.

Our method is significantly faster than a common non-hierarchical approach based on a complex Cartesian-product homogeneous model – up to ca. 10,000 times in the presented case studies based on dependable blocks configured as N-modular redundant system. Moreover, the proposed method is very accurate – the average relative error between hierarchical and classical solution based on Cartesian-product model is ca. 10^{-6} in the presented studies.

The speedup of the proposed method is most noticeable when the hierarchical model of the system can be constructed (the system is based on dependable blocks), but the method can be used to calculate a general non-homogeneous Markov chain. In the general case, the method will be slower than a common homogeneous approach, but it will be able to handle non-homogeneous parameters.

ACKNOWLEDGMENT

This research has been partially supported by the projects GA16-05179S, “Fault-Tolerant and Attack-Resistant Architectures Based on Programmable Devices: Research of Interplay and Common Features”, OP VVV funded project CZ.02.1.01/0.0/0.0/16_019/0000765 “Research Center for Informatics”, and Czech Technical University grant SGS17/213/OHK3/3T/18.

REFERENCES

- [1] M. Kohlík, “Hierarchical Dependability Models Based on Markov Chains”, Dissertation thesis, Czech Technical University in Prague, 2015. Available from: https://ddd.fit.cvut.cz/PhD/PhDThesis_Kohlík.pdf
- [2] “Electronic Reliability Design Handbook – MIL-HDBK-338B”, US Department of Defense, 1998. Available from: http://www.weibull.com/mil_std/mil_hdbk_338b.pdf
- [3] Eric W. Weisstein, “Chapman-Kolmogorov Equation”, From MathWorld—A Wolfram Web Resource. <http://mathworld.wolfram.com/Chapman-KolmogorovEquation.html>
- [4] R. Ocaña-Rilola, “Two Methods To Estimate Homogenous Markov Processes”, Journal of Modern Applied Statistical Methods: Vol. 1, Iss. 1, Article 17, 2002.
- [5] R. Ocaña-Rilola, “Non-homogeneous Markov Processes for Biomedical Data Analysis”, Biometrical Journal: Vol. 47, Iss. 3, 2005.
- [6] V. Vais, S. Racek, “Experimental evaluation of regular events occurrence in continuous-time markov models”, In Proceedings of the Eleventh International Conference on Informatics, Košice, Slovakia, 2011, pp. 143–146.
- [7] Wolfram. Mathematica. 2019. Available from: <http://www.wolfram.com/mathematica/>
- [8] R. Dobiáš, H. Kubátová, “The Common 2oo2 Safety Model for Signalling and Interlocking Equipments”, In *Electronic Circuits and Systems Conference*, Slovak Univ. of Technology, 2005, pp. 81–84.
- [9] R. Dobiáš, H. Kubátová, “FPGA based design of the railway’s interlocking equipments”, In *Digital System Design, 2004. DSD 2004. Euromicro Symposium on*, Aug 2004, pp. 467–473.